AVR32769: How to Compile the standalone AVR32 Software Framework in AVR32 Studio V2

1. Introduction

The purpose of this application note is to show how to compile any of the application and driver examples provided in the AVR®32 Software Framework using the AVR32 Studio® V2.

AVR®32 UC3 Microcontrollers

AVR32 Application Note

32115A–AVR32–05/09

2. Requirements

This application note requires that AVR32 Studio V2 be installed on your PC.

3. Reference

- AVR32 Studio V2 home page
  http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116
- AVR32 Software Framework home page
  http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4192

4. Overview

4.1 AVR32 Studio V2

4.1.1 Standard Make vs. Managed Make

Here is a discussion on which kind of project to create using the AVR32 Studio V2.

4.1.1.1 Managed Make Project

This is typically a kind of project that a developer must use when starting a new project from scratch. In this case, makefile creation will be transparent and GCC tools will be automatically handled.

- Manages compiles and tool-chain directly
- No makefile edition
- Fine control over compile, link settings

4.1.1.2 Standard Make Project

This is a kind of project for advanced users that want to reuse or create their own makefile.

- Re-uses existing makefiles
- Simple integration with arbitrary build systems
- Parsing of tool-chain output to generate error markers

Note: This kind of project will be used in this application note since all the AVR32 Software Framework examples are delivered with their own 'ready to use' GCC makefiles.
No modification of these makefiles is expected from the user.

4.2 AVR32 Software Framework

Install the standalone version of the AVR32 Software Framework package to you hard drive. AVR32 Software Framework can be downloaded from the Atmel web site at:

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4192

If needed, update the GCC header files by those contained in the AVR32 Software Framework:

1. Go to the `\UTILS\AVR32_HEADER_FILES` folder of the AVR32 Software Framework.
2. Unzip the AVR32_Header_Files.zip file in the `C:\Program Files\Atmel\AVR Tools\AVR32 Toolchain\avr32\include` folder.
3. Answer yes to all for header files replacement.

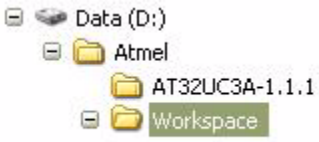
Note: The drive letter may change depending on your directory install.

5. Project Creation with AVR32 Studio V2

The following is a step-by-step procedure showing how to compile any AVR32 UC3 application example.

Note: In this example the workspace and the AVR32 Software Framework are stored in `D:\Atmel`

1. Launch AVR32 Studio.
2. Create a new workspace.
   – File menu ?Switch Workspace and enter the workspace place and name: e.g. `D:\Atmel\Workspace`
3. Workspace is placed at the same level as the root folder of the AVR32 Software Framework: `\AT32UC3n-x.y.z`



   Note: For proper operation, the AVR32 workspace must always be placed outside of the AVR32 Software Framework folder.
4. Create a Standard Make project:
   – File menu ?New -> Other -> C -> AVR32 C Project (Make)
   – Fill the AVR32 C Project (Make) window with the project name, and target MCU. usage.
5. Link the AVR32 Software Framework files to your project

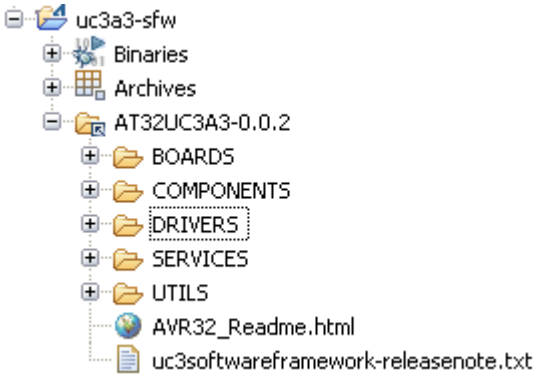
   Note: Contrary to what an “Import” command does (file or directory copy in the project folder) here the files are added by reference. Thus, all the compilation will be done inside the AVR32 Software Framework folder.
   The target MCU field should be set according to the software framework used.

   – File menu ?> New ?> Other
   – In the “Select a wizard” window select General ?> Folder then click Next
   – In the “Folder” window click on Advanced >>

– Check the “Link to folder in the file system” check box and browse to the AVR32 Software Framework folder:e.g. \AT32UC3A-x.y.z

– Click Finish

The project pane should now look like this:

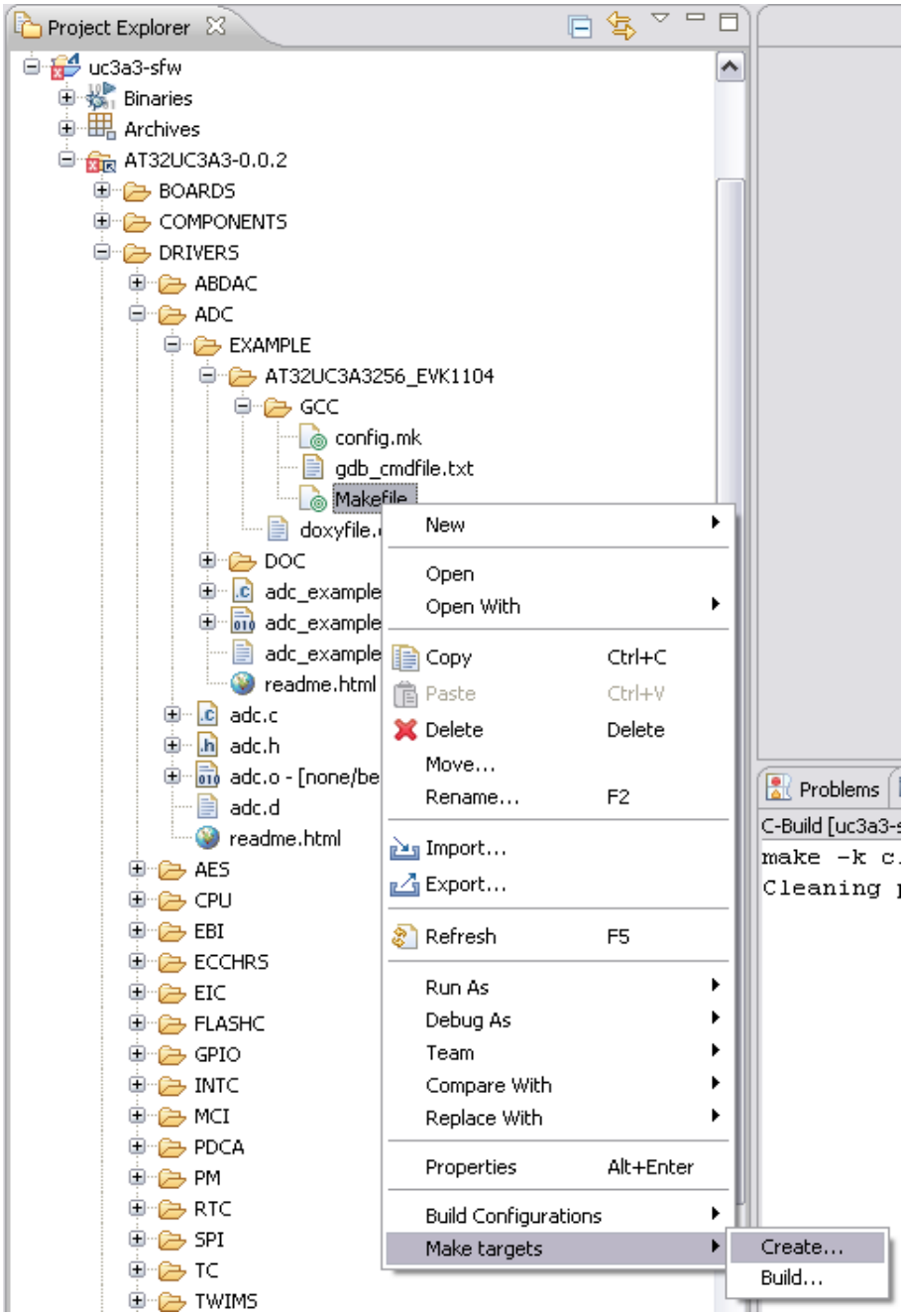


Note the arrow on the folder icon showing a linked folder.

6. Create the Make targets
   This will create the different targets that you will use for compiling,

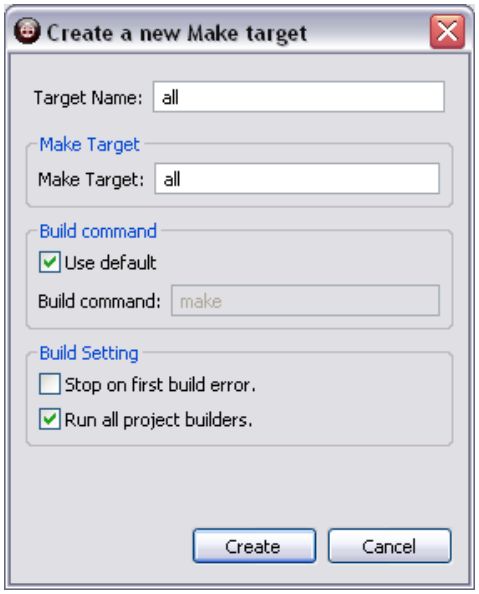
– In the project pane, browse any example to the Makefile folder: eg. for the ADC example: `\DRIVERS\ADC\EXAMPLE\AT32UC3Ax_EVK110x\GCC`

– Right-click on the Makefile in the `GCC` folder and select the 'Make Target'-> Create item:



Note: For proper operation, it is important to create the make target from the folder where the makefile is located. The makefile of examples inside the AVR32 Software Framework are always located inside a GCC folder.

– Fill the ‘Target Name:’ field, and fill the ‘Make Target:’ field with the make goal then click Create. Refer to Table 5-1 to get a list of all available options.
Here is an example allowing to compile the ‘Control Panel’ application:



Note: To easily sort the targets, it is recommended to name the target starting by the application or driver example name: e.g. Panel, TC-1, ADC…

Table 5-1. Make Goal List

Make Goal		Description
[all]		Default goal: build the project (update)
clean		Clean up the project
rebuild		Rebuild the project (clean + all)
ccversion		Display CC version information
cppfiles	file.i	Generate preprocessed files from C source files
asfiles	file.x	Generate preprocessed assembler files from C and assembler source files
objfiles	file.o	Generate object files from C and assembler source files
a	file.a	Archive: create A output file from object files
elf	file.elf	Link: create ELF output file from object files
lss	file.lss	Create extended listing from target output file
sym	file.sym	Create symbol table from target output file
sizes		Display target size information
isp		Use ISP instead of JTAGICE mkII when programming
cpuinfo		Get CPU information
halt		Stop CPU execution
chiperase		Perform a JTAG Chip Erase command
erase		Perform a flash chip erase
program		Program MCU memory from ELF output file

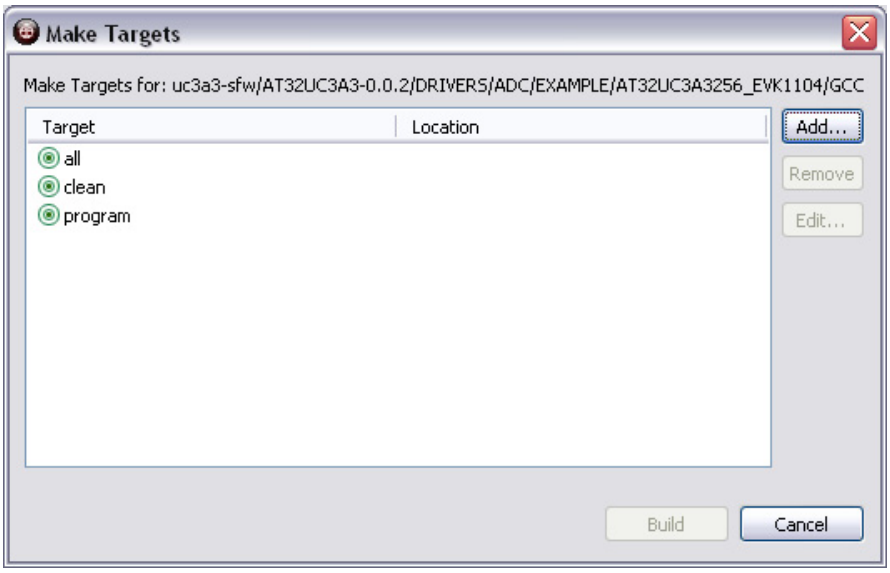
Table 5-1. Make Goal List (Continued)

Make Goal	Description
secureflash	Protect chip by setting security bit
reset	Reset MCU
debug	Open a debug connection with the MCU
run	Start CPU execution
readregs	Read CPU registers
verbose	Display main executed commands

These target goals can be combined to create complex targets. For example:

- “rebuild program run” goals
  will clean and compile the project, program the target through the JTAGICE mkII and launch the program execution.
- “rebuild isp program run” goals
  will clean and compile the project, program the target through the ISP bootloader and launch the program execution.

7. Build the Make targets
   – Right click anywhere in the project pane and select the ‘Make target -> Build’ item. The “Make Targets” window contains all the created targets. Here is an example including make targets for the ADC example :



   – Click Build to launch the highlighted make target.

   Note: This window allows also to manage the targets: adding, removing, editing. Take care to highlight a GCC folder in the project pane before adding a new target.

Note: Application and drivers example targets are sharing some resources that may be compiled using different options. In order to avoid any link errors due to a previous compilation, it is recommended to first clean a target before launching a first build. Thus old object files are deleted.

8. To launch a debug session:
   a. configure the target window: The easiest way of adding a target is to use the "Scan Targets" action. Open the "AVR32 Targets view" and right-click in it. This will bring up the view context menu. Select the action and let it discover whatever devices you have plugged into the USB port. As long as you have not specified a target board a red sign with a white line across (one-way sign) will be visible for the target. If no target was discovered you can manually add a new one by pressing the "Add target" tool button in the upper right corner in the view. The Target view contains the list of targets, and allows the user to perform actions on the targets as well as set up the targets. A target consists of three main components: adapter, board and microcontroller unit (MCU). The adapter role, is often performed by a JTAGICE mkII that communicates with the MCU and is responsible for executing the programming and/or debugging commands. The board is the hardware where the MCU is mounted, and may contain external memory which can be used by the MCU. Now open the "Properties" view (if not visible in the main window by default go to “Window -> Show View -> Properties) and select the target that was just discovered. Because the “Properties view” is shared by other functions or values of AVR32 Studio another click on the target may be needed to display its properties in the pane. If the target was added manually you must select the "Adapter" tab and set the correct adapter. Otherwise the adapter should be correct. Go to the "Board" tab and select the board you have. The MCU will be automatically selected if there is only

one MCU available for that board. Otherwise you must also select the correct MCU.

b. Right click on the generated .elf file in the /GCC folder. Select Debug As -> AVR32 Application. Errors existing in the projects may appear, selectg continue. A pop up “confirm perspective switch” may appear, select yes.

c. The debug session is launched.

6. Atmel Technical Support Center

Atmel has several support channels available. We encourage you to register and use our web portal for several reasons:

- All your requests are managed in one place. Easy both to submit new request, and get the follow-up on old requests.
- FAQ Access. We provide a large FAQ-database through our web site.
- You can apply for Free Newsletters on AVR32.

Support channels:

- Web: http://support.atmel.no/
- E-mail: avr32@atmel.com

Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature